The Hardware and the Network: Total-System Strategies for Power Aware Wireless Microsensors

Rex Min, Manish Bhardwaj, Nathan Ickes, Alice Wang, Anantha Chandrakasan
Massachusetts Institute of Technology, Cambridge, MA
{rmin, manishb, nickes, aliwang, anantha}@mtl.mit.edu

Abstract

The small battery capacity, ubiquity, and operational diversity of wireless microsensor networks create unprecedented energy management challenges. The energy consumption of microsensors is determined not only by the node's physical hardware, but also by the algorithms and protocols that impart functional demands on the hardware. We therefore present design methodologies that foster energy savings through collaboration across the hardware, algorithm, and network layers, in contrast to techniques that have explored only one of these spaces. For instance, dynamic voltage scaling is coupled with the intelligent distribution of within-network computation to extend latency deadlines and decrease supply voltages. The quality of communication is parameterized into metrics that drive the performance and energy consumption of the communication subsystem. Finally, the energy consumption of radios is carefully characterized to improve the efficiency of multihop routes. Collaboration between node software and hardware, and among the distributed nodes of the network, improve energy-efficiency and extend operational lifetime.

1. Introduction

The idea of wireless microsensor networks has garnered a great deal of attention and interest. A distributed wireless microsensor network [1] consists of hundreds to several thousands of small sensor nodes scattered throughout an area of interest. Each node individually monitors its environment and collects data as directed by the user, while the network collaborates as a whole to deliver high-quality observations to a central base station. The fusion of observations from different perspectives offers a high-resolution, multi-dimensional picture of the environment that is not possible with fewer sensors. The sheer number of nodes naturally leads to the network's fault-tolerance and robustness to the loss of individual nodes, making maintenance unnecessary. Nodes can be deployed simply by scattering them about the region of interest or dropping them by air; the nodes can organize themselves into networks without user intervention.

These advantages, as well as the nodes' small size, make sensor networks ideal for any number of inhospitable or inaccessible locations where deployment is difficult, wires impractical, and maintenance impossible. For instance, microsensors deployed in hostile environments can monitor climate, classify moving vehicles, or provide an early warning of chemical or radiation hazards. A microsensor network distributed around the body, or perhaps within it, can offer a rich fusion of vital signs to medical professionals. Sensors within machines such as copiers and industrial robots, can detect and report emerging faults without the usual tangle of wires [2, 3, 4]. Microsensor networks promise to revolutionize how data is gathered.

A microsensor node integrates sensing, processing, and communication sub-systems. Several researchers have demonstrated operational nodes with low-power commercial, off-the-shelf (COTS) components [5, 6, 7]. A representative example is depicted in Figure 1 , which contains an on-board acoustic sensor and A/D, a StrongARM processor for data and protocol processing, power regulators for dynamic energy management, and a 2.4 GHz Bluetooth compatible radio. This particular node, dubbed µAMPS-1 [8], integrates these components onto stackable 55mm x 55mm boards as illustrated by Figure 2. Energy dissipation of this COTS-based microsensor node is reduced through a variety of techniques including fine-

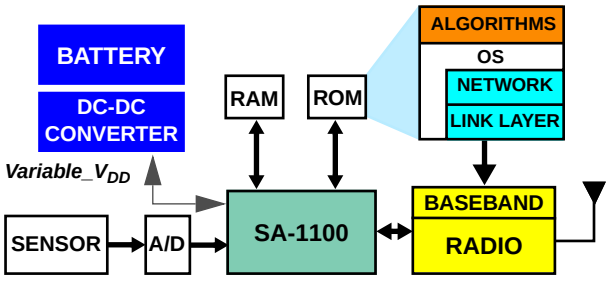

Figure 1: Architectural overview of the first generation MIT sensor node (The MIT µAMPS project) based on low-power off-the-shelf components. The node allows algorithms to gracefully scale its energy consumption by modifying hardware parameters.

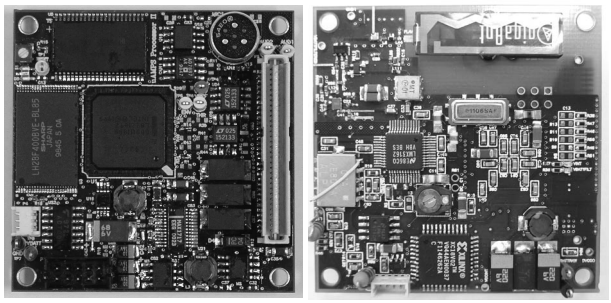

Figure 2: The µAMPS-1 sensor node: sensor and processor sub-system (left) and radio communication module (right).

grain shutdown of inactive components through operating system control, dynamic voltage and frequency scaling of the processor core, approximate-processing algorithms, and adjustable radio transmission power based on the required range. More than one order of magnitude variation in power dissipation is exhibited among the different power management states. Power consumption can range from 30 mW to over 1 W.

2. The Hardware: Energy Scalability

Power aware hardware reacts gracefully to constantly changing operational demands. As performance demands increase or decrease, power aware hardware scales energy consumption accordingly to adjust its performance on-the-fly. Graceful energy scalability is highly desirable for any energy-constrained wireless node since the operational demands on a real-world node constantly change, and the peak performance of the node is rarely needed. Energy scalability is effected by key parameters that act as knobs to adjust energy and performance simultaneously. The following characterization of a microsensor node's energy consumption reveals a number of these knobs within the processing and radio subsystems.

2.1 Processing Energy

The energy of digital circuits within the node can be characterized as the energy required to process a bit of data, whether that processing be a filtering algorithm, beamforming operation, or Viterbi decoding. The energy consumed per bit is expressed as the sum of digital switching and leakage energies:

$$E_{bit}(V_{DD}, C_{bit}, T_{bit}) = C_{bit}V_{DD}^2 + \left(V_{DD}I_0 10^{\frac{-V_{TH}}{S}}\right)T_{bit} \quad (1)$$

V_{DD} the supply voltage, C_{bit} is the switched capacitance per processed bit, and T_{bit} the computational time required per bit. C_{bit} and T_{bit} are themselves functions of the hardware implementation and the operation being performed. I_0 and S, which model digital leakage current, are functions of the process technology. V_{TH} is the threshold voltage of the transistors.

Energy-saving opportunities in digital processing can be gleaned from the parameters of Eq. (1). Varying V_{DD} through dynamic voltage scaling reduces switching energy quadratically. Emerging techniques for dynamic V_{TH} scaling reduce the exponential leakage term. Finally, the amount of computation per bit (represented by C_{bit}) can be modulated by varying the strength of an algorithm.

Dynamic voltage scaling (DVS) reduces active power by varying the supply voltage and clock frequency depending on the computational load [8, 9]. The processor's frequency is reduced to the lowest possible level that meets the required performance constraints, and the supply voltage is reduced to attain additional energy savings from the lower frequency. DVS is highly applicable to processors within sensor nodes, as processor load can vary significantly based on the node's operational mode (e.g., sensing/processing vs. data relay) and event statistics. Figure 3 depicts the measured energy consumed per operation for the StrongARM processor with respect to the processor clock frequency and supply voltage. The graph illustrates the advantage of voltage scaling at reduced processing loads. Note that at a fixed supply voltage, the leakage energy per operation increases as the allowed switching time per operation increases. The supply voltage is scheduled by an embedded operating system and is controlled at the physical-layer by an efficient variable voltage DC-DC converter.

Further energy reductions are possible by coordinating a second parameter, the device threshold voltage V_{TH}, with the supply voltage. Device thresholds can be adjusted through substrate biasing in triple-well CMOS technology. Just as V_{DD} scaling exploits the trade-off between propaga-

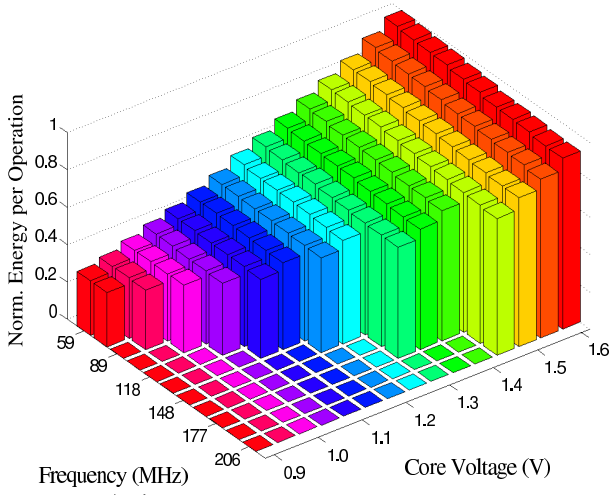

Figure 3: Dynamic Voltage Scaling on the SA-1100.

tion delay and switching energy, V_{TH} scaling can exploit a trade-off between propagation delay and sub-threshold leakage power. For every clock frequency, there exists an optimal pair of parameters (V_{TH}, V_{DD}) that minimizes energy.

Figure 4 illustrates the impact of varying the power supply and threshold voltage on the performance and energy dissipation of submicron circuits. Diagonal lines indicate the (V_{TH}, V_{DD}) values that will support operation of a 16-bit adder for widely varying levels of performance between 10 kHz and 100 MHz. Circular contours represent the increasing amount of energy consumed for each (V_{TH}, V_{DD}) around a minimum value at (460 mV, 300 mV). The minimum-energy operating points for each clock frequency, then, occur where each frequency plot is tangent to an energy contour. These optimal points are joined by a dotted line that represents the optimal (V_{TH}, V_{DD}) for each clock frequency.

The optimal selections are most intuitive at the extremes of performance. The highest performance can only be achieved with a low V_{TH} and high V_{DD}, at the expense of high switching and leakage energies. In the kilohertz regime where high circuit latencies are tolerable, not only is V_{DD} lowered for reduced switching energy, but V_{TH} is simultaneously raised to suppress leakage. In fact, the supply voltage is scaled *below* the threshold voltage; load capacitances are switched by subthreshold leakage currents. As leakage currents are orders of magnitude lower than drain currents in the strong inversion regime, both performance and active power dissipation are substantially reduced. Given the low performance demands of microsensor nodes, operation in the subthreshold operating regime is an exciting possibility.

Finally, the amount of workload imposed on the digital circuits (represented by C_{bit}) can be varied by altering the

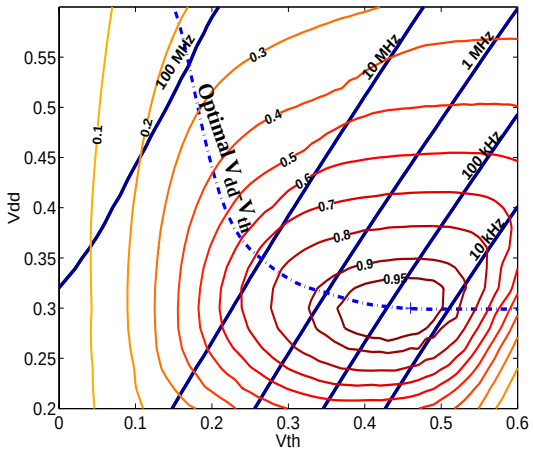

Figure 4: Average energy dissipation and performance for a 16-bit adder. The dotted line indicates the adder's optimal (V_{th}, V_{dd}) operating points for varying clock frequencies.

strength of a processing algorithm. For instance, in a Viterbi decoder, computation varies exponentially with the constraint length of the convolutional code. Workload variation can also be exploited by power-aware algorithms that gracefully degrade result accuracy as the number of operations (and hence the required energy) is scaled back. Power aware algorithms offer incremental refinement: the algorithm yields a reasonable approximation of the answer after a minimal number of operations, and the result grows increasingly accurate as more and more operations are performed.

2.2 Radio Energy

The energy of radio communication is the sum of the transmission energy of the sending node and the receiver energy of the receiving node. A comprehensive model for transmission energy is

$$E_{tx}(P_{rad}, N) = P_{start}T_{start} + \frac{N}{R}[P_{txElec} + (\alpha_{rad} + \beta_{rad}P_{rad})] \quad (2)$$

The two main terms in the expression represent the energies of startup and transmission respectively. P_{start} and T_{start} represent the power and latency of radio startup, P_{txElec} the active transmission power, P_{rad} the power radiated from the antenna, N the number of bits transmitted, and R the radio bit rate. The terms α_{rad} and β_{rad} allow a linearized model for power amplifier and end-to-end antenna losses. The efficiency of the power amplifier is often overlooked in energy models, but with typical efficiencies of around 10%, it is a crucial consideration for energy-efficient wireless communication.

Airborne radio transmissions are attenuated by a path loss in a power-law with distance, meaning that the relationship between radiated power P_{rad} and distance d is

$$P_{rad}(P_{rcvd}, d) = P_{rcvd}P_{1m}d^n \quad (3)$$

P_{rcvd} is the power incident at the receiver, which may be, for instance, the receiver's sensitivity. P_{1m} is the signal attenuation at one meter from the transmitter.

The energy of the receiver is modeled as

$$E_{rx}(N) = P_{start}T_{start} + \frac{N}{R}P_{rxElec} + NE_{decbit} \quad (4)$$

Many terms are repeated from above; P_{rxElec} represents the active receive power, and E_{decbit} the energy per bit required to decode any error correction that may have been applied. Note that this last term is a digital processing energy in the form of Eq. (1).

A simple hook for energy *vs.* quality scalability within the radio is the variation of the radiated power P_{rad}, which allows the range or error rate of communication to be tied to the energy allocated to transmission. Note that the varia-

tion of a convolutional code's constraint length, a scalability knob discussed in Section 2.1, expands the number of transmitted bits *N*. Hence, this knob has energy implications for both digital and radio energy.

2.3 Power Scalability of µAMPS-1

Many of the preceding "knobs" for energy scalability are incorporated into the processor and radio modules of the µAMPS-1 sensor node (Figure 2 above). The dynamic voltage scaled processor operates from 0.9V to 1.6V at clock frequencies ranging from 59 to 220 MHz. As shown in Figure 5, the processor consumes up to three times less energy when performing a computation at lower frequencies and voltages. The µAMPS-1 radio transmitter offers a power amplifier with variable output power between +0 dBm and +20 dBm. As shown in Figure 6, this two order of magnitude range in output power translates into a factor of five in hardware energy consumption (200 mW to 1 W).

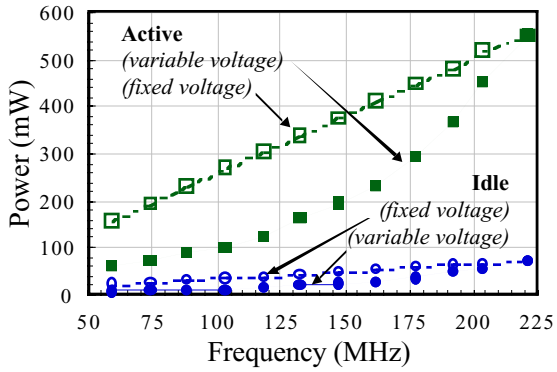

Figure 5: Dynamic voltage scaling on the µAMPS-1 node can reduce processing energy by nearly a factor of three.

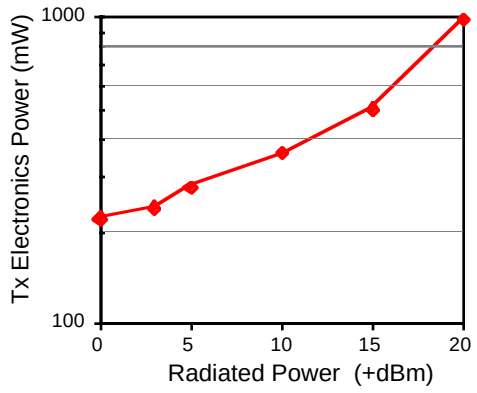

Figure 6: The quality and power consumption of the µAMPS-1 radio transmitter can be scaled by adjusting the output power between +0 dBm and +20 dBm.

3. The Network: Collaborative Strategies

Inter-node communication provides opportunities for two forms of energy-saving collaboration. Collaboration can occur in two senses: nodes may cooperate among each other to reduce energy consumed across the network, and communication protocols may cooperate with the hardware on which they run to reduce within-node energy consumption.

3.1 Distributed Processing

Algorithm implementations for a sensor network can take advantage of the network's inherent capability for parallel processing to further reduce energy. Partitioning a computation among multiple sensor nodes and performing the computation in parallel permits a greater allowable latency per computation, allowing energy savings through frequency and voltage scaling.

As an example, consider a target tracking application that requires sensor data to be transformed into the frequency domain through 1024-point FFTs [9]. The FFT results are phase-shifted and summed in a frequency-domain beamformer to calculate signal energies in 12 uniform directions, and the line-of-bearing (LOB) is estimated as the direction with the most signal energy. By intersecting multiple LOB's at the basestation, the source's location can be determined. Figure 7a demonstrates the tracking application performed with traditional clustering techniques for a 7 sensor cluster. The sensors (S1-S6) collect data and transmit the data directly to the cluster-head (S7), where the FFT, beamforming and LOB estimation are performed. Measurements on the SA-1100 at an operating voltage of 1.5V and frequency of 206 MHz show that the tracking application dissipates 27.27 mJ of energy.

Distributing the FFT computation among the sensors reduces energy dissipation. In the distributed processing scenario of Figure 7b, the sensors collect data and perform the FFTs before transmitting the FFT results to the cluster-head. At the cluster-head, the FFT results are beamformed

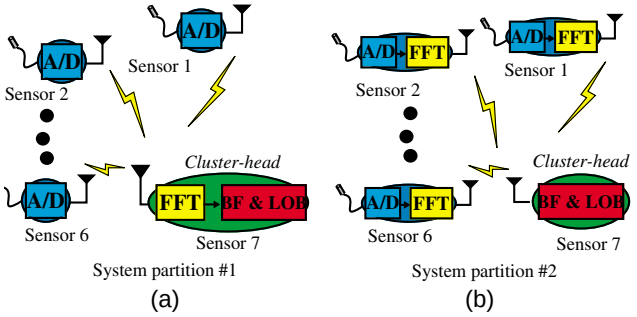

Figure 7: Two system partitions for dividing computation across a network. In (a), all computation is performed at a single node. In (b), computation is divided among nodes.

and the LOB estimate is found. Since the 7 FFTs are computed in parallel, we can reduce the supply voltage and frequency without sacrificing latency. When the FFTs are performed at 0.9V, and the beamforming and LOB estimation at the cluster-head are performed at 1.3V, then the tracking application dissipates 15.16 mJ, a 44% improvement in energy dissipation.

3.2 Energy-Scalable Communication

Graceful energy *vs.* quality scalability for wireless communication can be achieved once the notion of communication "quality" is defined. Hence, we define communication quality by four of its fundamental metrics: *range*, *latency*, *reliability*, and *energy*. We then introduce an API that allows an application to specify bounds on these metrics. The latter three metrics can be bounded by direct specification:

- `set_max_latency(double usecs)`
- `set_min_reliability(double ber)`
- `set_max_energy(double ujoules)`

With cooperation from a protocol layer that maintains approximate distances to—and numbers of—neighboring nodes, the communication range desired by an application can be expressed in a variety of ways, whichever is most convenient to the application:

- `set_range(Distance d)`
- `set_range(int numberOfNearestNeighbors)`
- `set_destination(Node n)`
- `set_destination(Nodes n[])`

This skeletal communication API allows an application to expose tolerable bounds on latency, reliability, range, and energy. Application designers that utilize wireless communication have historically been reluctant to manipulate hardware energy knobs such as processor voltages or transmit power. Introducing proper abstractions between communication software and hardware can therefore encourage energy savings.

To bridge the gap between these performance parameters and the actual hardware "knobs" for energy scalability, we introduce a power-aware middleware layer as illustrated in Figure 8. The middleware layer is empowered with accurate hardware energy models for the node's digital processing circuits and radio transceiver, such as those presented in Section 2. This layer also maintains models for the latency, reliability, and range of communication as a function of the low-level hardware parameters.

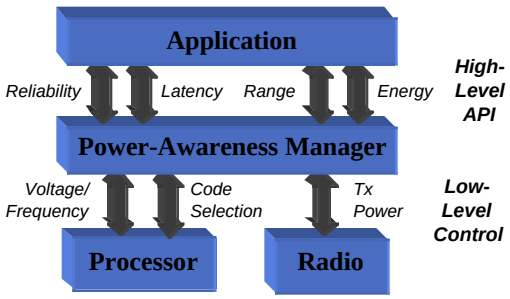

Figure 8: A power-aware interface bridges an application's quality requests for communication and the hardware's energy scalability.

With this information, the middleware can select the minimum-energy hardware settings for the performance level commanded through the API [10]. Figure 9 illustrates a sample operational policy for the µAMPS-1 node, in which an API-specified reliability and range (x- and y-axes) are mapped to the radio transmission power and convolutional coding scheme that result in minimum energy consumption. The upper-left hand corner of the graph represents the lowest quality transmission, achieved using uncoded communication at a low transmission power. As quality demands increase (moving toward the lower-right of the graph), µAMPS-1 first commands more transmit power. As µAMPS-1 has a high energy cost associated with Viterbi decoding, increasing quality through transmission power is more energy-efficient than utilizing a convolutional code. Convolutional coding is finally used once the quality demands exceed the capabilities of uncoded, high-power communication. The choice of hardware policies (coding *versus* radiation in this example) is unique to the energy consumption characteristics of each hardware platform.

The amount of energy consumed by the operational policies of Figure 9 is illustrated by Figure 10. This figure illustrates the energy required to attain a specified range and reliability by choosing the minimum-energy hardware policies for each operating point. We have achieved energy *vs.* quality scaling in communication through a middleware layer that serves as both an abstraction boundary and an intelligent point of control.

3.3 Power Aware Multihop Routing

Since the path loss of radio transmission scales with distance in a greater-than-linear fashion, transmission energy

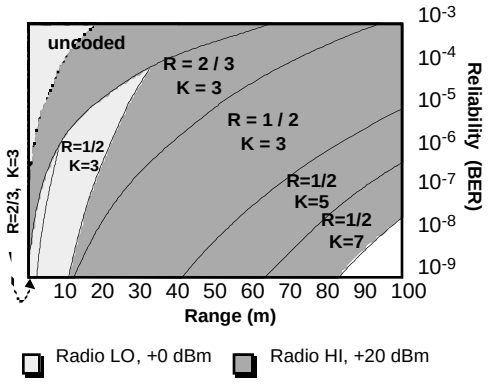

Figure 9: Least-energy hardware policy for single-hop communication given a specified reliability and range, considering both transmit and receive energy. N = 1000.

can be reduced by dividing a long transmission into several shorter ones. Intermediate nodes between a data source and destination can serve as relays that receive and rebroadcast data. This concept, known as *multihop* communication [11], is analogous to the use of buffers over a long, on-chip interconnect.

Figure 11 illustrates multihop communication to a base station across a distance d using h hops. The power consumed by this communication $P(h, d)$ is

$$P(h,d) = (h-1)P_{rxElec} + h\left[P_{txRad1}\left(\frac{d}{h}\right)^{r} + P_{txElec}\right] \quad (5)$$

where P_{rxElec} and P_{txElec} represent the power required by the receive and transmit electronics, P_{txRad1} is the radiated power required for a successful one-meter transmission, and r is the path loss exponent. Since the last hop is always received by an energy-unconstrained base station, there are h transmitting and h-1 receiving nodes. Figure 12 evaluates equation Eq. (5) over varying total transmission distances d and one to four hops, using representative power consumption parameters for COTS radios.

The introduction of relay nodes is clearly a balancing act between reduced transmission energy and increased receive energy. Hops that are too short lead to excessive receive energy. Hops that are too long lead to excessive path loss. In between these extremes is an optimum transmission distance called the *characteristic distance* d_{char} [12]. The characteristic distance depends only on the energy consumption of the hardware and the path loss coefficient; d_{char} alone determines the optimal number of hops. For typical COTS-based sensor nodes, d_{char} is about 35 meters.

The existence of a characteristic distance has two practical implications for microsensor networks. First, there are a large class of applications for which the entire network diameter will be less than d_{char}. For these applications, the best communication policy is not to employ multihop at all; direct transmission from each node to the base station is the most energy-efficient communication scheme. For today's radio hardware, the typical d_{char} of 35 meters exceeds the size of many interior spaces. Hence, until advances in low-power receiver technology lead to a reduction in d_{char}, most indoor microsensor networks will not save energy using a multihop routing protocol.

Our second concern is that it is often impractical to ensure that all nodes are spaced exactly d_{char} apart. Nodes may dropped by air, or their deployment constrained by terrain or physical obstacles. Suppose that the deployed nodes are placed as in Figure 13, a line of nodes and a base station separated a distance of either d_{char} or $d_{\text{char}}/2$. As the figure illustrates, there are three possible multi-hop policies from the farthest node to the base station.

With the knowledge that minimum-energy hops have a distance d_{char}, we can naively choose the first policy in Fig-

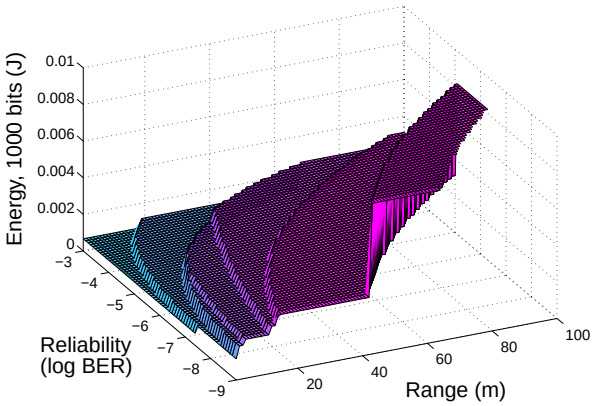

Figure 10: Energy-scalable behavior of µAMPS-1 node for the middleware-selected operational policies of Figure 9.

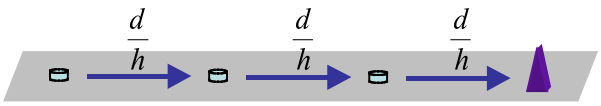

Figure 11: Multihop routing.

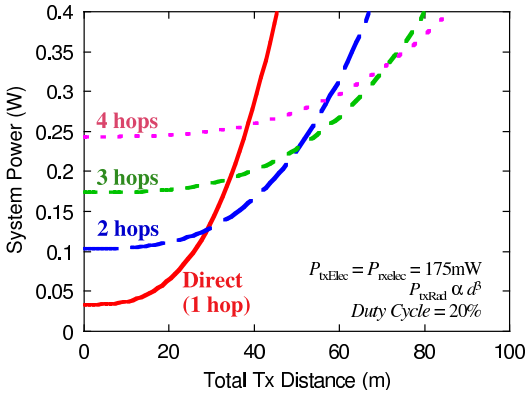

Figure 12: As the distance between a source node and base station increases, the number of intermediate hops required for maximal energy-efficiency increases.

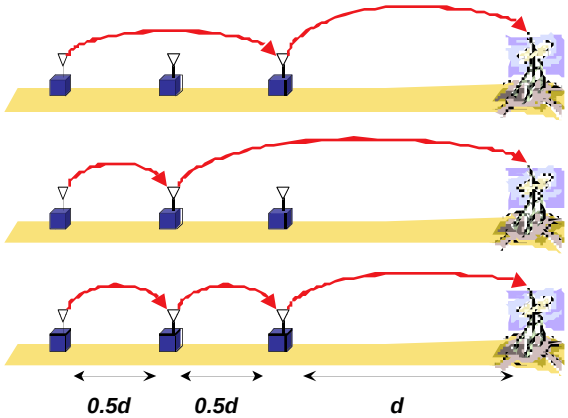

Figure 13: Three multihop policies are possible when two intermediate nodes stand between the transmitting sensor and base station.

ure 13, which specifies two hops of equal distance d_{char}. Doing so, however, ignores the presence of an additional relay node whose energy is available for consumption. The other two policies in Figure 13 include this relay node, but each of these would cause the node with the longest hop to run out of energy well before the others have been exhausted. The problem is one of an *uneven spatial distribution* of energy.

In cases like these, the solution for maximum relay lifetime turns out to be a rotation of roles over time, leading to the depletion of all nodes at about the same time. The final numerical result is a function of the node spacing (in terms of characteristic distance) and the initial node energies, and can be solved using a linear program that considers these constraints [12]. For this example, the optimal role assignment for equal-energy nodes utilizes sends 3/11 of all packets through each of the one-hop routes and the remaining 5/11 of packets through the two-hop route. Role rotation compensates for uneven spatial distribution in energy.

4. Conclusion

Wireless microsensor networks utilize hundreds to thousands of tiny, inexpensive, and densely placed nodes to achieve unprecedented sensing resolution and fault-tolerance. The nodes' limited energy capacity and extended lifetime requirements demand that all aspects of a microsensor node be designed in a power aware fashion. Both digital and analog node circuitry must reflect the distinctive operational characteristics of microsensors, notably performance demands that have a low average but high variability. Power aware design cannot end with hardware; software and communication protocols for microsensors must actively contribute to energy savings through energy-efficient operational policies and sacrifices in performance. This necessarily implies that the software, or a suitable layer of abstraction, must call upon accurate energy and performance models for the hardware upon which it runs. As with all emerging wireless applications, the future of microsensor networks is reliant on designs that successfully meld unique operational demands with innovative circuit design for maximal energy efficiency.

Acknowledgments

The authors wish to thank Kevin Atkinson, Fred Lee, and Piyada Phanaphat for their work with μAMPS-1; and Steven V. Kosonocky for close collaboration on dynamic threshold scaled circuits.

This research is sponsored by the Defense Advanced Research Project Agency (DARPA) Power Aware Computing/Communication Program and the Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0551; by the Army Research Laboratory (ARL) Collaborative Technology Alliance, through BAE Systems, Inc. subcontract RK7854; and by Hewlett-Packard under the HP/MIT Alliance. Rex Min is an NDSEG Fellow. Alice Wang is an Intel Corporation Fellow.

References

[1] G. Asada *et al.*, "Wireless Integrated Network Sensors: Low Power Systems on a Chip," *IEEE ESSCIRC '98*, pp. 9-16, September 1998.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *ACM MobiCom '99*, pp. 263-270, August 1999.

[3] J. Kahn, R. Katz, and K. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," *ACM MobiCom '99*, pp. 271-278, August 1999.

[4] J. Rabaey *et al.*, "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design," *IEEE ISSCC*, pp. 200-201, July 2002.

[5] B. Warneke, B. Atwood, K.S.J. Pister, "Smart dust mote forerunners," Proc. 14th IEEE Int'l Conference on MEMS, Jan. 2001, pp. 357-360.

[6] J. Rabaey et al., "PicoRadio supports ad hoc ultra-low power wireless networking," Computer, vol. 33, no. 7, July 2000, pp. 42-48.

[7] G. Asada et al., "Wireless Integrated Network Sensors: low power systems on a chip," Proc. 1998 ESSCIRC, Sept. 1998.

[8] Shih, E. et al., "Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks", *Proc. MobiCOM 2001*, July 2001.

[9] A. Wang and A. Chandrakasan, "Energy Efficient System Partitioning for Distributed Wireless Sensor Networks", *Proc. ICASSP 2001*, May 2001

[10] R. Min and A. Chandrakasan, "A Framework for Energy-Scalable Communication in High-Density Wireless Networks," *Proc. ISLPED 2002*, pp. 36-41.

[11] E. Royer and C-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," IEEE Personal Communications, vol. 6, no. 2, April 1999, pp. 46-55.

[12] M. Bhardwaj, T. Garnett, and A. Chandrakasan, "Upper Bounds on the Lifetime of Sensor Networks," *Proc. ICC 2001*, vol. 3, June 2001, pp. 785-790.

[13] R. Min et al., "Low-Power Wireless Sensor Networks," *Proc. 14th Int'l Conf. on VLSI Design*, Bangalore, India, Jan. 2001, pp. 205-210.